A Latent Source Model for Online Time Series Classification*

George H. Chen
georgehc@mit.edu

Stanislav Nikolov
snikolov@twitter.com

Devavrat Shah
devavrat@mit.edu

Abstract

We study a binary classification problem whereby an infinite time series having one of two labels ("event" or "non-event") streams in, and we want to predict the label of the time series. Intuitively, the longer we wait, the more of the time series we see and so the more accurate our prediction could be. Conversely, making a prediction too early could result in a grossly inaccurate prediction. In numerous applications, such as predicting an imminent market crash or revealing which topics will go viral in a social network, making an accurate prediction as early as possible is highly valuable. Motivated by these applications, we propose a generative model for time series which we call a *latent source model* and which we use for non-parametric online time series classification. Our main assumption is that there are only a few ways in which a time series corresponds to an "event", such as a market crashing or a Twitter topic going viral, and that we have access to training data that are noisy versions of these few distinct modes. Our model naturally leads to weighted majority voting as a classification rule, which operates without knowing nor learning what the few latent sources are. We establish theoretical performance guarantees of weighted majority voting under the latent source model and then use the voting to predict which news topics on Twitter will go viral to become trends.

1 Introduction

Classifying time series is a widely studied problem. In this paper, we specifically consider the problem of time series classification in the "online" setting, where a time series is infinite in length and streams in. For simplicity, we assume that a single time series has just one binary label indicating whether the time series is an "event" or a "non-event". We use a running example from social media: consider a time series that tracks how much activity there is for a particular news topic. A question we could ask is "will this particular news topic go viral?" Here, a time series, which corresponds to a particular news topic, is an "event" if it goes viral at some point and is a "non-event" otherwise. We skirt the discussion of what makes a topic considered "viral" as this is irrelevant to our mathematical development.

Most similar to our work is that of Lee et al. [2012], who use a k-nearest-neighbor classifier to classify time series. Their work is experimental whereas our focus will largely be on developing some theoretical understanding for how well a nearest-neighbor-like weighted majority voting classifier performs under certain modeling assumptions, where an approximation of this weighted majority voting is a k-nearest-neighbor classifier. Even in the

*This work was supported in part by the Army Research Office under MURI Award 58153-MA-MUR.

"offline" setting in which the time series is finite in length and the entire series can be used for classification, a nearest-neighbor-based approach is hard to beat in terms of classification performance [Xi et al., 2006], working as well or better than time series classification approaches using, for example, neural networks [Nanopoulos et al., 2001], decision trees [Rodríguez and Alonso, 2004], or support vector machines [Wu and Chang, 2004]. However, to the best of our knowledge, there is no existing justification for such performance results of nearest-neighbor-like time series classifiers, specifically in understanding both when such approaches work well and how well. Lastly, we remark that handling the case where a single time series can have different labels at different times is beyond the scope of this work.

Our contributions. We present a generative model for time series which we call a *latent source model* and which we use for binary time series classification. In our model, time series are generated from a small collection of m unknown latent sources. We propose an accompanying classifier that corresponds to weighted majority voting, which compares the time series to be classified with each of the time series in the labeled training data; each training time series casts a vote in favor of its ground truth label. This voting is nonparametric in that it does not learn parameters for a model and is driven entirely by the training data. The unknown latent sources are never estimated; the training data serve as a proxy for these latent sources. We give an example of why estimating these latent sources under a parametric model could be problematic and result in a poor classifier. Under our latent source model, we show that as long as we have $\Theta(m \log m + m \log \frac{1}{\delta})$ time series in our training data, weighted majority voting correctly classifies a new time series with high probability ($\geq 1 - \delta$) after observing its first $O(\log m + \log 1/\delta)$ samples.

We formulate predicting which news topics on Twitter will go viral as an instance of our online time series classification problem. For simplicity, we take a news topic to have gone "viral" if Twitter has deemed it, at some point in time, a "trending topic".[1] We emphasize that our goal is *precognition* of trends: predicting whether a topic is going to be a trend before it is actually defined to be a trend by Twitter or, in theory, any other third party that we can collect ground truth labels from. Existing works that identify trends on Twitter [Becker et al., 2011, Cataldi et al., 2010, Mathioudakis and Koudas, 2010] instead, as part of their trend detection, define models for what trends are, which we do not do nor do we assume we have access to such definitions. In our experiments, weighted majority voting is able to predict whether a topic will be a trend in advance of Twitter 79% of the time, with a mean early advantage of 1 hour and 26 minutes and achieving a true positive rate of 95% and a false positive rate of 4%. The reason why a classifier as simple as weighted majority voting works so well in our experiments may be that there aren't actually that many ways in which a news topic on Twitter becomes a trend. We empirically find that the Twitter activity of a news topic that becomes a trend tends to follow one of a finite number of patterns, which could be thought of as latent sources. We don't know how many such patterns there are, but we hypothesize the number to be substantially smaller than the number of time series that we can collect for training data. Importantly, our experimental results suggest that a latent source model is a reasonable model for Twitter trend prediction.

Outline. We present weighted majority voting for online time series classification in

[1]While it is not public knowledge how Twitter defines a topic to be a trending topic, Twitter does provide information for which topics are trending topics. We take these labels to be ground truth, effectively treating how a topic goes viral to be a black box supplied by Twitter.

Section 2. We then provide our latent source model and theoretical performance guarantees of weighted majority voting under this model in Section 3. Experimental results for predicting which news topics on Twitter become trends are in Section 4. We conclude in Section 5.

Notation. Each time series is represented as a function mapping $\mathbb{Z}$ to $\mathbb{R}$, where we index time using $\mathbb{Z}$ for notationally convenience. However, we always assume time series to start at time step 1, meaning that any time series $q : \mathbb{Z} \to \mathbb{R}$ is 0 before time step 1: $q(t) = 0$ for all $t < 1$. For two time series $q, q' : \mathbb{Z} \to \mathbb{R}$ and integer time shift Δ, we use $q * \Delta$ to denote time series q advanced by Δ time steps, i.e., $(q * \Delta)(t) = q(t + \Delta)$ if $t + \Delta \geq 1$, and $q(t + \Delta) = 0$ otherwise. Finally, for positive integer window size T, we denote $\langle q, q' \rangle_T \triangleq \sum_{t=1}^{T} q(t)q'(t)$, and $\|q\|_T^2 \triangleq \langle q, q \rangle_T$.

2 Weighted Majority Voting

Suppose we have a time series $s : \mathbb{Z} \to \mathbb{R}$ that we want to classify as having either label $+1$ ("event") or -1 ("non-event"). We have access to labeled training data $\mathcal{R}_+$ and $\mathcal{R}_-$, which denote the sets of all time series with labels $+1$ and -1 respectively.

Each positively-labeled example $r \in \mathcal{R}_+$ casts a weighted vote $e^{-\gamma d^{(T)}(r,s)}$ on whether time series s has label $+1$, where $d^{(T)}(r, s)$ is some measure of similarity between the two time series r and s, superscript (T) indicates that we are only allowed to look at the first T time steps (i.e., time steps $1, 2, \ldots, T$) of s (but we're allowed to look outside of these time steps for the training time series r), and constant $\gamma > 0$ is a scaling parameter that determines the "sphere of influence" of each example. Similarly, each negatively-labeled example in $\mathcal{R}_-$ also casts a weighted vote on whether time series s has label -1. Essentially, each example time series in the training data says, with a certain confidence, "The observation looks like me, so it should have the same label as me."

For the similarity measure $d^{(T)}(r, s)$, we could, for example, use squared Euclidean distance between time series: $d^{(T)}(r, s) = \sum_{t=1}^{T}(r(t) - s(t))^2 = \|r - s\|_T^2$. However, this similarity measure only uses time steps $1, 2, \ldots, T$ of example time series r. Since time series in our training data are known, there is no reason why we should restrict our attention to their first T time steps. Thus, we use the following similarity measure:

$$d^{(T)}(r, s) = \min_{\Delta \in \{-\Delta_{\max}, \ldots, 0, \ldots, \Delta_{\max}\}} \sum_{t=1}^{T} (r(t + \Delta) - s(t))^2 = \min_{\Delta \in \{-\Delta_{\max}, \ldots, 0, \ldots, \Delta_{\max}\}} \|r * \Delta - s\|_T^2, \tag{1}$$

where we minimize over integer time shifts with a pre-specified maximum allowed time shift $\Delta_{\max} \geq 0$.

Finally, we sum up all of the weighted $+1$ votes and then all of the weighted -1 votes. The label with the majority of overall weighted votes is declared as the label for s:

$$L^{(T)}(s; \gamma) = \begin{cases} +1 & \text{if } \sum_{r \in \mathcal{R}_+} e^{-\gamma d^{(T)}(r,s)} \geq \sum_{r \in \mathcal{R}_-} e^{-\gamma d^{(T)}(r,s)}, \\ -1 & \text{otherwise.} \end{cases} \tag{2}$$

Using a larger time window size T corresponds to waiting longer before we make a prediction. We need to trade off how long we wait and how accurate we want our prediction. Also, note that k-nearest-neighbor classification corresponds to only considering the weighted votes from the k nearest neighbors among all training time series; all other votes are set to 0.

3 A Latent Source Model and Theoretical Guarantees

Before stating our main theoretical result, we provide our latent source model. We assume there to be m unknown latent sources (time series) that generate observed time series. Let $\mathcal{V}$ denote the set of all such latent sources; each latent source $v : \mathbb{Z} \to \mathbb{R}$ in $\mathcal{V}$ has a true label $+1$ or -1. Let $\mathcal{V}_+ \subset \mathcal{V}$ be the set of latent sources with label $+1$, and $\mathcal{V}_-$ be the set of those with label -1. Let $m_+ \triangleq |\mathcal{V}_+|$ and $m_- \triangleq |\mathcal{V}_-| = m - m_+$. The observed time series are generated from latent sources as follows:

1. Sample a latent source V from $\mathcal{V}$ uniformly at random. Let $L \in \{+1, -1\}$ be the label of V.
2. Sample integer time shift Δ uniformly from $\{0, 1, \dots \Delta_{\text{max}}\}$.
3. Output time series $S : \mathbb{Z} \to \mathbb{R}$ to be latent source V advanced by Δ time steps, followed by adding noise signal $E : \mathbb{N} \to \mathbb{R}$, i.e., $S(t) = V(t + \Delta) + E(t)$ for $t \geq 1$ (by convention, $S(t) = 0$ for $t < 1$). We describe the noise model for E later. The label associated with the generated time series S is the same as that of V, i.e., L.

The above generative process defines our latent source model.

Importantly, we make no assumptions about the structure of the latent sources. For instance, the latent sources could be tiled as shown in Figure 1, where they are evenly separated vertically and alternate between the two different classes $+1$ and -1. With a parametric model like a k-component Gaussian mixture model, estimating these latent sources could be problematic. For example, if we take any two adjacent latent sources with label $+1$ and cluster them, then this cluster could be confused with the latent source having label -1 that is sandwiched in between. As the latent sources are not actually known, we would only see noisy versions of them, which only complicates estimating what they are. In this example, the k-component Gaussian mixture model needed for label $+1$ would require k to be the exact number of latent sources with label $+1$, which is unknown. As we discuss

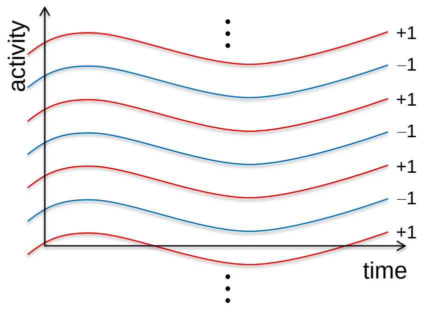


Figure 1: Example of latent sources superimposed, where each latent source is shifted vertically in amplitude such that every other latent source has label $+1$ and the rest have label -1.

next, for inference, we shall sidestep learning the latent sources altogether, instead using training data as a proxy for latent sources.

Inference. If we knew the latent sources and if noise entries $E(t)$ were i.i.d. $\mathcal{N}(0, \frac{1}{2\gamma})$ across t, then the *maximum a posteriori* (MAP) estimate for label L given an observed time series $S = s$ is

$$\widehat{L}^{(T)}_{\text{MAP}}(s;\gamma) = \begin{cases} +1 & \text{if } \Lambda^{(T)}_{\text{MAP}}(s;\gamma) \geq 1, \\ -1 & \text{otherwise,} \end{cases} \tag{3}$$

where

$$\Lambda^{(T)}_{\text{MAP}}(s;\gamma) \triangleq \frac{\sum_{v_+ \in \mathcal{V}_+} \sum_{\Delta_+ \in \mathcal{D}_+} \exp\big(-\gamma\|v_+ * \Delta_+ - s\|^2_T\big)}{\sum_{v_- \in \mathcal{V}_-} \sum_{\Delta_- \in \mathcal{D}_+} \exp\big(-\gamma\|v_- * \Delta_- - s\|^2_T\big)}, \tag{4}$$

and $\mathcal{D}_+ \triangleq \{0, \ldots, \Delta_{\max}\}$.

However, we do not know the latent sources, nor do we know if the noise is i.i.d. Gaussian. We assume that we have access to training data as given in Section 2. We make a further assumption that the training data were sampled out of the latent source model and that we have n different training time series. Denote $n_+ \triangleq |\mathcal{R}_+|$, $n_- \triangleq |\mathcal{R}_-|$, $\mathcal{R} \triangleq \mathcal{R}_+ \cup \mathcal{R}_-$, and $\mathcal{D} \triangleq \{-\Delta_{\max}, \ldots, 0, \ldots, \Delta_{\max}\}$. Then we approximate the MAP classifier by using training data as a proxy for the latent sources. Specifically, we take ratio (4), replace the inner sum by a minimum in the exponent, replace $\mathcal{V}_+$ and $\mathcal{V}_-$ by $\mathcal{R}_+$ and $\mathcal{R}_-$, and replace $\mathcal{D}_+$ by $\mathcal{D}$ to obtain the ratio:

$$\Lambda^{(T)}(s;\gamma) \triangleq \frac{\sum_{r_+ \in \mathcal{R}_+} \exp\big(-\gamma\big(\min_{\Delta_+ \in \mathcal{D}} \|r_+ * \Delta_+ - s\|^2_T\big)\big)}{\sum_{r_- \in \mathcal{R}_-} \exp\big(-\gamma\big(\min_{\Delta_- \in \mathcal{D}} \|r_- * \Delta_- - s\|^2_T\big)\big)}. \tag{5}$$

Plugging $\Lambda^{(T)}$ in place of $\Lambda^{(T)}_{\text{MAP}}$ in classification rule (3) yields the weighted majority voting rule (2) from Section 2. Note that weighted majority voting could be interpreted as a *smoothed* nearest-neighbor approximation whereby we only consider the time-shifted version of each example time series that is closest to the observed time series s.

Lastly, applications may call for trading off true and false positive rates. One way to do so is to generalize the decision rule (3) to declare the label to be $+1$ if $\Lambda^{(T)}(s,\gamma) \geq \theta$ and then sweep over different values of parameter $\theta > 0$. The resulting decision rule, which we refer to as *generalized weighted majority voting*, is thus:

$$L^{(T)}_{\theta}(s;\gamma) = \begin{cases} +1 & \text{if } \Lambda^{(T)}(s,\gamma) \geq \theta, \\ -1 & \text{otherwise,} \end{cases} \tag{6}$$

where setting $\theta = 1$ recovers the usual weighted majority voting (2). This modification to the classifier can be thought of as adjusting the priors on the relative sizes of the two classes. Our theoretical results to follow actually cover this more general case rather than only that of $\theta = 1$.

Main result. We now state the main theoretical result of this paper which identifies conditions under which generalized weighted majority voting (6) can classify the time series correctly. Concretely, the conditions state how large the training data set needs to be in order to achieve high classification accuracy under our generative model with a reasonably generic noise model. In what follows, we shall assume that the noise time series $E : \mathbb{Z} \to \mathbb{R}$

is such that $E(t)$ are i.i.d. across t and the distribution of $E(1)$ (or $E(t)$ for any t) is zero mean, *sub-Gaussian* with standard-deviation parameter σ. That is, for any $\lambda \in \mathbb{R}$,

$$\mathbb{E}\big[\exp\big(\lambda E(1)\big)\big] \le \exp\Big(\frac{1}{2}\lambda^2\sigma^2\Big). \tag{7}$$

Note that the family of sub-Gaussian distributions includes a variety of distributions, such as a zero-mean Gaussian with standard deviation σ and a uniform distribution over $[-\sigma, \sigma]$.

Next, we define a notion of the "gap" between $\mathcal{R}_+$ and $\mathcal{R}_-$ restricted to time length T and time shift $\Delta_{\max}$ as follows:

$$G^{(T)}(\mathcal{R}_+, \mathcal{R}_-, \Delta_{\max}) \triangleq \min_{\substack{r_+ \in \mathcal{R}_+, r_- \in \mathcal{R}_-, \\ \Delta_+, \Delta_- \in \mathcal{D}}} \|r_+ * \Delta_+ - r_- * \Delta_-\|_T^2. \tag{8}$$

This quantity measures how far apart the two different classes are if we only look at length-T chunks of each time series and allow all shifts of at most $\Delta_{\max}$ time steps in either direction. Then our main theoretical result is as follows:

Theorem 1 *Under the above described setup and with $n \ge \beta m \log m + 1$ time series in the training data for a pre-specified $\beta > 0$, the probability of misclassification using generalized weighted majority voting $\widehat{L}_\theta^{(T)}(\cdot; \gamma)$ satisfies the bound*

$$\begin{aligned}&\mathbb{P}(\textit{misclassification of } S \textit{ using its first } T \textit{ samples}) \\ &\qquad \le \Big(\theta + \frac{1}{\theta}\Big)(2\Delta_{\max}+1) n \exp\big(-(\gamma - 4\sigma^2\gamma^2) G^{(T)}(\mathcal{R}_+, \mathcal{R}_-, \Delta_{\max})\big) + m^{-\beta+1}.\end{aligned} \tag{9}$$

We defer the proof of Theorem 1 to Section 3.1 and instead turn now to the theorem's implications. Given error tolerance $\delta \in (0,1)$ and with choice $\gamma \in (0, \frac{1}{4\sigma^2})$, then upper bound (9) is at most δ (by having each of the two terms on the right-hand side be $\le \frac{\delta}{2}$) if $n \ge m \log m + m \log \frac{2}{\delta} + 1$, and

$$G^{(T)}(\mathcal{R}_+, \mathcal{R}_-, \Delta_{\max}) \ge \frac{\log(\theta + \frac{1}{\theta}) + \log(2\Delta_{\max}+1) + \log n + \log\frac{2}{\delta}}{\gamma - 4\sigma^2\gamma^2}. \tag{10}$$

In particular, if we have access to a large enough pool of labeled time series, i.e., the pool has $\Omega(m \log m + m \log \frac{1}{\delta})$ time series, then we can subsample $n = \Theta(m \log m + m \log \frac{1}{\delta})$ of them for use as training data. With this training data and with choice $\gamma = \frac{1}{8\sigma^2}$, generalized weighted majority voting (6) correctly classifies a new time series S with probability at least $1 - \delta$ if[2]

$$G^{(T)}(\mathcal{R}_+, \mathcal{R}_-, \Delta_{\max}) = \Omega\left(\frac{\log(\theta + \frac{1}{\theta}) + \log(2\Delta_{\max}+1) + \log m + \log\frac{1}{\delta}}{\gamma}\right). \tag{11}$$

That is, the effective gap between sets $\mathcal{R}_+$ and $\mathcal{R}_-$ needs to grow only logarithmic in the number of distinct latent sources m in order for weighted majority voting to be able to classify correctly with high probability. Now, assuming that the original unknown latent sources are separated (otherwise, there is no hope to distinguish between the classes using any classifier) and the gap (squared distance) grows linearly with T (in a sense, this is necessary even to combat the noise), then observing the first $O(\log m + \log 1/\delta)$ samples from the time series is sufficient to classify it correctly with probability at least $1 - \delta$.

[2] Note that $\log(m(\log m + \log\frac{1}{\delta})) + \log\frac{1}{\delta} = \log m + \log(\log m + \log\frac{1}{\delta}) + \log\frac{1}{\delta} = \Theta(\log m + \log\frac{1}{\delta})$.

3.1 Proof of Theorem 1

Let S be the time series with an unknown label that we wish to classify using training data. As per the model, there exists a latent source V, shift $\Delta' \in \mathcal{D}$, and noise signal E' such that

$$S = V * \Delta' + E'. \tag{12}$$

With a training set of size $n \geq \beta m \log m + 1$, for each latent source $V \in \mathcal{V}$, there exists at least one signal in $\mathcal{R}$ that is generated from V with probability at least $1 - m^{-\beta+1}$ (coupon collector's problem). Henceforth, we assume that this event holds.

Note that R is generated from V as

$$R = V * \Delta'' + E'', \tag{13}$$

where $\Delta' \in \mathcal{D}$ and E' is noise signal, independent of E. Therefore, we can rewrite S in the form of R as follows:

$$S = R * \Delta + E \tag{14}$$

where $\Delta = \Delta' - \Delta'' \in \mathcal{D}$ and $E = E' - E'' * \Delta$. Since E' and E'' are i.i.d. over time and sub-Gaussian with parameter σ, one can easily verify that E is i.i.d. over time and sub-Gaussian with parameter $\sqrt{2}\sigma$.

We now bound the probability of error of classifier $L^{(T)}(\cdot;\gamma)$. The probability of error or misclassification using the first T samples of S is given by

$$\begin{aligned}&\mathbb{P}\big(\text{misclassification of } S \text{ using its first } T \text{ samples}\big)\\ &= \mathbb{P}(\widehat{L}_\theta^{(T)}(S;\gamma) = -1 | L = +1)\underbrace{\mathbb{P}(L = +1)}_{m_+/m} + \mathbb{P}(\widehat{L}_\theta^{(T)}(S;\gamma) = +1 | L = -1)\underbrace{\mathbb{P}(L = -1)}_{m_-/m}.\end{aligned} \tag{15}$$

In the remainder of the proof, we primarily show how to bound $\mathbb{P}(\widehat{L}_\theta^{(T)}(S;\gamma) = -1|L = +1)$. The bound for $\mathbb{P}(\widehat{L}_\theta^{(T)}(S;\gamma) = +1|L = -1)$ is almost identical. By Markov's inequality,

$$\mathbb{P}(\widehat{L}_\theta^{(T)}(S;\gamma) = -1|L = +1) = \mathbb{P}\left(\frac{1}{\Lambda^{(T)}(S;\gamma)} \geq \frac{1}{\theta}\Big| L = +1\right) \leq \theta\mathbb{E}\left[\frac{1}{\Lambda^{(T)}(S;\gamma)}\Big| L = +1\right]. \tag{16}$$

Now,

$$\mathbb{E}\left[\frac{1}{\Lambda^{(T)}(S;\gamma)}\Big| L = +1\right] \leq \max_{r_+ \in \mathcal{R}_+, \Delta_+ \in \mathcal{D}} \mathbb{E}_E\left[\frac{1}{\Lambda^{(T)}(r_+ * \Delta_+ + E;\gamma)}\right]. \tag{17}$$

With the above inequality in mind, we next bound $1/\Lambda^{(T)}(r_+ + E;\gamma)$. Let $r_+ \in \mathcal{R}$ and $\Delta_+ \in \mathcal{D}$. Then for any time series s,

$$\Lambda^{(T)}(s;\gamma) \geq \frac{\exp\big(-\gamma\|r_+ * \Delta_+ - s\|_T^2\big)}{\max_{r_- \in \mathcal{R}_-, \Delta_- \in \mathcal{D}} \exp\big(-\gamma\|r_- * \Delta_- - s\|_T^2\big)}. \tag{18}$$

After evaluating the above for $s = r_+ * \Delta_+ + E$, a bit of algebra shows that

$$\begin{aligned}&\frac{1}{\Lambda^{(T)}(r_+ * \Delta_+ + E;\gamma)}\\ &\leq \max_{r_- \in \mathcal{R}_-, \Delta_- \in \mathcal{D}} \Big\{\exp\big(-\gamma\|r_+ * \Delta_+ - r_- * \Delta_-\|_T^2\big)\exp\big(-2\gamma\langle r_+ * \Delta_+ - r_- * \Delta_-, E\rangle_T\big)\Big\}.\end{aligned} \tag{19}$$

Using (19), we obtain the following bound:

$$
\begin{aligned}
&\mathbb{E}_E\left[\frac{1}{\Lambda^{(T)}(r_+ * \Delta_+ + E;\gamma)}\right] \\
&\le \mathbb{E}_E\Bigg[\max_{\substack{r_-\in\mathcal{R}_-,\\ \Delta_-\in\mathcal{D}}}\Big\{\exp\big(-\gamma\|r_+ * \Delta_+ - r_- * \Delta_-\|_T^2\big)\exp\big(-2\gamma\langle r_+ * \Delta_+ - r_- * \Delta_-, E\rangle_T\big)\Big\}\Bigg] \\
&\le \mathbb{E}_E\Bigg[\sum_{\substack{r_-\in\mathcal{R}_-,\\ \Delta_-\in\mathcal{D}}}\Big\{\exp\big(-\gamma\|r_+ * \Delta_+ - r_- * \Delta_-\|_T^2\big)\exp\big(-2\gamma\langle r_+ * \Delta_+ - r_- * \Delta_-, E\rangle_T\big)\Big\}\Bigg] \\
&\overset{(i)}{=} \sum_{\substack{r_-\in\mathcal{R}_-,\\ \Delta_-\in\mathcal{D}}}\exp\big(-\gamma\|r_+ * \Delta_+ - r_- * \Delta_-\|_T^2\big)\prod_{t=1}^{T}\mathbb{E}_{E(t)}\big[\exp\big(-2\gamma(r_+(t+\Delta_+) - r_-(t+\Delta_-))E(t)\big)\big] \\
&\overset{(ii)}{\le} \sum_{\substack{r_-\in\mathcal{R}_-,\\ \Delta_-\in\mathcal{D}}}\exp\big(-\gamma\|r_+ * \Delta_+ - r_- * \Delta_-\|_T^2\big)\prod_{t=1}^{T}\exp\big(4\sigma^2\gamma^2(r_+(t+\Delta_+) - r_-(t+\Delta_-))^2\big) \\
&= \sum_{\substack{r_-\in\mathcal{R}_-,\\ \Delta_-\in\mathcal{D}}}\exp\big(-(\gamma-4\sigma^2\gamma^2)\|r_+ * \Delta_+ - r_- * \Delta_-\|_T^2\big) \\
&\le (2\Delta_{\max}+1)n_-\exp\big(-(\gamma-4\sigma^2\gamma^2)G^{(T)}\big), \qquad (20)
\end{aligned}
$$

where step (i) uses independence of entries of E, step (ii) uses sub-Gaussianity, and the last line abbreviates $G^{(T)} \equiv G^{(T)}(\mathcal{R}_+, \mathcal{R}_-, \Delta_{\max})$. Stringing together inequalities (16), (17), and (20), we obtain

$$\mathbb{P}(\widehat{L}_\theta^{(T)}(S;\gamma) = -1 | L = +1) \le \theta(2\Delta_{\max}+1)n_-\exp\big(-(\gamma-4\sigma^2\gamma^2)G^{(T)}\big). \qquad (21)$$

Repeating a similar argument yields

$$\mathbb{P}(\widehat{L}_\theta^{(T)}(S;\gamma) = +1 | L = -1) \le \frac{1}{\theta}(2\Delta_{\max}+1)n_+\exp\big(-(\gamma-4\sigma^2\gamma^2)G^{(T)}\big). \qquad (22)$$

Finally, plugging (21) and (22) into (15) followed by applying inequalities $n_+, n_- \le n$ and $m_+, m_- \le m$ gives

$$
\begin{aligned}
&\mathbb{P}\big(\text{misclassification of } S \text{ using its first } T \text{ samples}\big) \\
&\quad \le \Big(\theta + \frac{1}{\theta}\Big)(2\Delta_{\max}+1)n\exp\big(-(\gamma-4\sigma^2\gamma^2)G^{(T)}\big). \qquad (23)
\end{aligned}
$$

This completes the proof of Theorem 1.

4 Detecting Trends on Twitter

We now formulate predicting which news topics will go viral on Twitter as an instance of our online time series classification setup and show some experimental results. As some background, Twitter is a real-time global communication network whose users can post short public messages called *Tweets*, which are then broadcast to a user's followers. Often, emerging topics of interest are discussed on Twitter in real time. Inevitably, certain topics gain sudden popularity and — in Twitter speak — begin to *trend*. Twitter surfaces such topics as a list of top ten *trending topics*, or *trends*. For example, a sudden rise in Tweets mentioning "Barclays" (such as "*Barclays fined $450 million for manipulating Libor interest rate*") may cause the topic "Barclays" to trend. We wish to detect when a topic starts trending, ideally as early as possible and with a low rate of error (false detections and false non-detections).

Data. We sampled 500 examples of trends at random from a list of June 2012 news trends and recorded the earliest time each topic trended within this month. Before sampling, we filtered out trends that never achieved a rank of 3 or better on the Twitter trends list[3] as well as trends that lasted for less than 30 minutes as to keep our trend examples reasonably salient. We also sampled 500 examples of non-trends at random from a list of n-grams (of sizes 1, 2, and 3) appearing in Tweets created in June 2012, where we filter out any n-gram containing words that appeared in one of our 500 chosen trend examples. Note that as we do not know how Twitter chooses what phrases are considered as topic phrases (and are candidates for trending topics), it's unclear what the size of the non-trending topics category is in comparison to the size of the trending topics category. Thus, for simplicity, we intentionally control for the class sizes by setting them equal. In practice, one could still expressly assemble the training data to have pre-specified class sizes and then tune θ for generalized weighted majority voting (6). In our experiments, we just use the usual weighted majority voting (2) to classify time series.

From these examples of trends and non-trends, we then created example time series of activity for trends and non-trends. The time series of activity for a topic is based on the rate of Tweets about that topic over time. To approximate this rate, we gathered 10% of all Tweets from June 2012, placed them into two-minute buckets according to their timestamps, and counted the number of Tweets in each bucket. We denote the count at the t-th time bucket as $\rho(t)$, which we refer to as the raw rate.

To create example time series, we transform the raw rate in a number of ways based on our observations of trending and non-trending activity. We observed that trending activity is characterized by spikes above some baseline rate, whereas non-trending activity has fewer, if any spikes. For example, a non-trending topic such as "city" has a very high, but mostly constant rate because it is a common word. In contrast, soon-to-be-trending topics like "Miss USA" will initially have a low rate, but will also have bursts in activity as the news spreads. To emphasize the parts of the rate signal above the baseline and de-emphasize the parts below the baseline, we define a baseline-normalized signal $\rho_b(t) \triangleq \rho(t)/\sum_{\tau=1}^{t}\rho(\tau)$.

A related observation is that the Tweet rate for a trending topic typically contains larger and more sudden spikes than those of non-trending topics. We reward such spikes by emphasizing them, while de-emphasizing smaller spikes. To do so, we define a baseline-and-spike-normalized rate $\rho_{b,s}(t) \triangleq |\rho_b(t)-\rho_b(t-1)|^\alpha$ in terms of the already baseline-normalized rate ρ_b; parameter $\alpha \geq 1$ controls how much spikes are rewarded (we used $\alpha = 1.2$). In addition, we convolve the result with a smoothing window to eliminate noise and effectively

[3]On Twitter, topics trending topics compete for the top ten spots whereas we are only detecting whether a topic will trend or not.

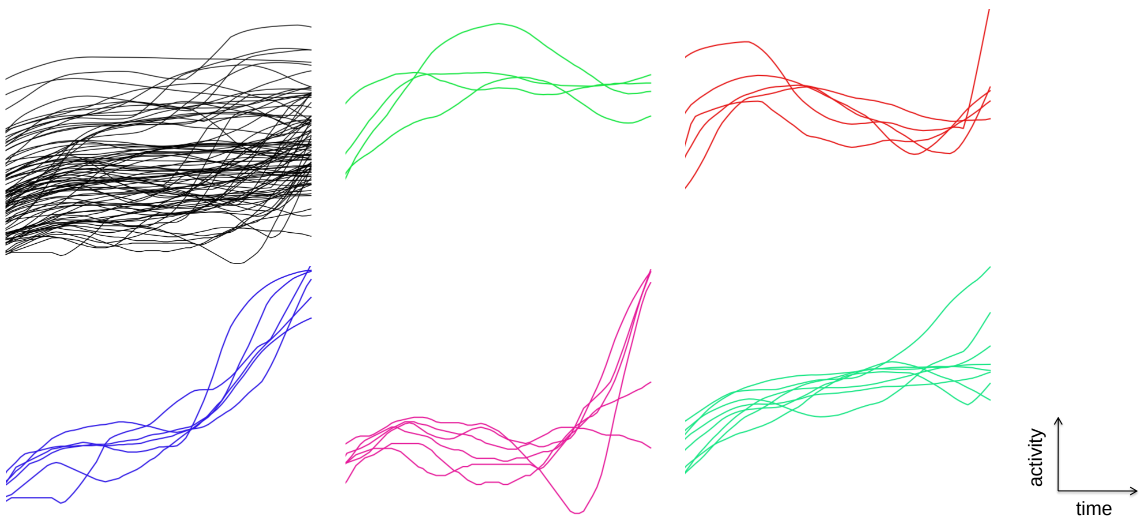


Figure 2: The top left shows example patterns of activity leading up to a news topic becoming trending. These patterns superimposed look like clutter, but we can separate the patterns into different clusters, as shown in the next five clusters of patterns. Each cluster represents a "way" that a news topic can become trending.

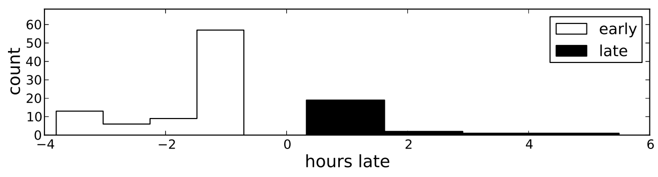
measure the volume of Tweets in a sliding window of length T_{smooth}:

$$\rho_{b,s,c}(t) \triangleq \sum_{\tau=t-T_{smooth}+1}^{t} \rho_{b,s}(\tau). \tag{24}$$

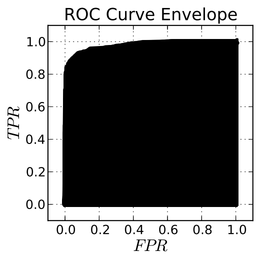
Finally, the spread of a topic from person to person can be thought of as a branching process in which a population of users "affected" by a topic grows exponentially with time, with the exponent depending on the details of the model [Asur et al., 2011]. This intuition suggests using a logarithmic scaling for the volume of Tweets: $\rho_{b,s,c,l}(t) \triangleq \log \rho_{b,s,c}(t)$.

The time series $\rho_{b,s,c,l}$ contains data from the entire window in which data was collected. To construct the sets of example time series $\mathcal{R}_+$ and $\mathcal{R}_-$, we keep only a small h-hour slice of representative activity r for each topic. Namely, each of the final example time series r used in the training data is truncated to only contain the h hours of activity in the corresponding transformed time series $\rho_{b,s,c,l}$. For time series corresponding to trending topics, these h hours are taken from the time leading up to when the topic was first declared by Twitter to be trending. For time series corresponding to non-trending topics, the h-hour window of activity is sampled at random from all the activity for the topic. We empirically found that how news topics become trending tend to follow a finite number of patterns; we show a few examples of these patterns in Figure 2.

Experiment. For a given parameter setting, we divided the set of trends and non-trends into a training set (50%) and test set (50%) at random and performed detection on the test set using the example time series in the training set. Trend detection was performed using weighted majority voting (2) on activity in a window of $2h$ hours, centered at the trend onset for trends, and sampled randomly for non-trends. We restrict detection to this time window to avoid detecting earlier times that a topic became trending, if it trended



(a)



(b)

Figure 3: (a) Weighted majority voting achieves a low error rate (FPR = 0.04, TPR = 0.95) and detects trending topics in advance of Twitter 79% of the time, with a mean of 1.43 hours; parameters: $\gamma = 10, T = 115, T_{smooth} = 80, h = 7$. (b) Envelope of all ROC curves shows the between false positives and true positives.

multiple times. We then measured the false positive rate, true positive rate, and the time of detection (if any). For trends, we computed how early or late the detection was compared to the true trend onset. We explored the following parameters: h, the length in hours of each example time series; T, the number of (initial) samples in the observed time series s that we use for classification; γ, the scaling parameter; T_{smooth}, the width of the smoothing window. In all cases, constant $\Delta_{\max}$ in the decision rule (2) is set to be the maximum possible, i.e., since observed signal s has T samples, we compare s with all T-sized chunks of each example time series r in training data.

Results. For a range of parameters, we are able to detect trending topics before they appear on Twitter's trending topics list. Figure 3a shows that for the given parameter setting, we are able to detect trending topics before Twitter does 79% of the time, and when we do, we detect them an average of 1.43 hours earlier. Furthermore, we achieve a low error rate: a true positive rate of 95% and a false positive rate of just 4%. Naturally, there are tradeoffs between false positive rate (FPR), true positive rate (TPR), and relative detection time that depend on the choice of parameters. An aggressive parameter setting will yield early detection and a high TPR, but only at the expense of a high FPR. A conservative parameter setting will result in a low FPR, but only at the expense of late detection and a low TPR. An in-between setting will strive for the right balance. We show this tradeoff in two ways. First, by varying a single parameter and keeping the rest fixed, we generated an ROC curve that describes the tradeoff between FPR and TPR. Figure 3b shows the envelope of all ROC curves, which can be interpreted as the best "achievable" ROC curve. Second, we broke the results up by where they fall on the ROC curve — top ("aggressive"), bottom ("conservative"), and center ("in-between") — and showed the distribution of early and late relative detection times for each (Figure 4).

We discuss some fine details of the experimental setup. Due to restrictions on the Twitter data available, while we could determine whether a trending topic is categorized as news based on user-curated lists of "news" people on Twitter, we did not have such labels for individual Tweets. Thus, the example time series that we use as training data contain Tweets that are both news and non-news. We also reran our experiments using only non-news Tweets and found similar results except that we do not detect trends as early as

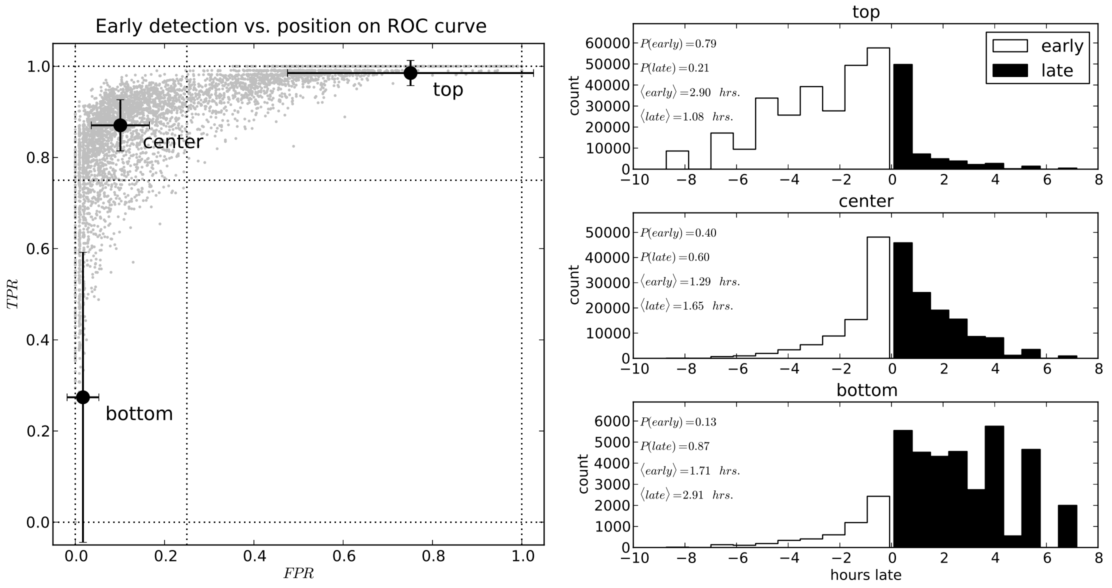


Figure 4: Distribution of early and late relative detection times for "aggressive" (top), "conservative" (bottom) and "in-between" (center) parameter settings.

before; however, weighted majority voting still detects trends in advance of Twitter 79% of the time.

5 Conclusion

We proposed a generative model for time series called a latent source model and provided theoretical guarantees of weighted majority voting under this model. Our experiments show that despite its simplicity, weighted majority voting can often detect trends on Twitter earlier than Twitter itself. These results, along with our observation that how a news topic becomes a trend tends to follow only some number of canonical patterns, suggest that our latent source model is a reasonable model for Twitter trend prediction. We expect that, while remaining agnostic to the definition of a trend, achieving more remarkable trend forecasts with classifiers that wisely leverage domain knowledge should be possible. For time series analysis more generally, future directions include extending our theoretical analysis to other problems such as detecting whether a time series is anomalous or predicting its future values.

References

Sitaram Asur, Bernardo A. Huberman, Gábor Szabó, and Chunyan Wang. Trends in social media: Persistence and decay. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.

Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on Twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.

Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, 2010.

Yen-Hsien Lee, Chih-Ping Wei, Tsang-Hsiang Cheng, and Ching-Ting Yang. Nearest-neighbor-based approach to time-series classification. *Decision Support Systems*, 53(1): 207 – 217, 2012.

Michael Mathioudakis and Nick Koudas. Twittermonitor: trend detection over the Twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010.

Alex Nanopoulos, Rob Alcock, and Yannis Manolopoulos. Feature-based classification of time-series data. *International Journal of Computer Research*, 10, 2001.

Juan J. Rodríguez and Carlos J. Alonso. Interval and dynamic time warping-based decision trees. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, 2004.

Yi Wu and Edward Y. Chang. Distance-function design and fusion for sequence data. In *Proceedings of the 2004 ACM International Conference on Information and Knowledge Management*, 2004.

Xiaopeng Xi, Eamonn J. Keogh, Christian R. Shelton, Li Wei, and Chotirat Ann Ratanamahatana. Fast time series classification using numerosity reduction. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, 2006.